REMARKS

Claims 1-31 are pending in the application. Claims 1-31 are rejected.

Claims 9, 19, and 29-31 have been cancelled herein.

Claims 29-31 are rejected under 35 U.S.C.101 because the invention set forth in these claims is directed to non-statutory subject matter. Claims 29-31 have been cancelled obviating the rejection.

Apfel et al. (5,974,454) (hereinafter "Apfel")

Claims 1, 8-10, 21 28, 29 and 31 are rejected under 35 U.S.C. 102 as being anticipated by Apfel. Claims 9, 29 and 31 are cancelled obviating the rejection.

It is respectfully submitted that the structure of Apfel is basically different from that of the claimed invention. In backbone networks, it is very difficult to stop the entire system or a process executing in the system to update software and avoid service interruptions. Therefore, programs must be dynamically replaced during regular operations of the communication networks. The dynamic linkage recited in the present claimed invention makes it possible to replace programs without service interruptions. The features recited in applicant's claimed invention are not disclosed by Apfel.

For example applicant's claim 1 recites: a program memory unit which is allocated to an activated process, and temporarily stores at least one program component transferred from said program storing unit and a program executing unit which dynamically links one of said one or more program components corresponding to said version information contained in said message received by said message receiving unit, to said program memory unit, so as to enable execution of said one of said one or more program components in said process.

Summary of Apfel (col. 2, lines 10-67 and Abstract)

Apfel teaches a method of updating a software program module component stored on a computer or installing new software program module components over a distributed computer network, such as the Internet.

Apfel describes if the current date (a date of installing a program) is on or after a date stored in a registry key on a computer, a database query is sent from the computer over a computer network, such as the Internet, to a database server.

At the database server, in response to receiving the database query, determination is made whether an upgrade package for the software program module component is available. If an upgrade package for the software program module component is available, then an upgrade package message is sent from the database server to the computer.

At the computer, in response to receiving the upgrade package message, a determination is made whether a user wants to download the upgrade package. If the user wants to download it, then an upgrade package query is sent by the computer via computer networks to a package server.

At the package server, in response to receiving the upgrade package query, the upgrade package is retrieved and sent over computer networks to the computer. The upgrade package is installed on the computer.

From the above structure, an upgraded software program is distributed via a network, such as Internet, and is input automatically to a user's computer.

Difference between claim 1 and Apfel

The feature of applicant's claim 1 is that a program executing unit dynamically links one of new program components to a program memory unit in which another program is running, so

as to enable execution of the new program components in a process. These features are supported in applicant's specification for example see Fig. 1 and p. 23, line 6- p. 26, line 7.

Therefore from applicant's recited features a new program is transferred to memory without stopping a transaction (a sequence of processes necessary for executing the corresponding program) of a currently running process. As described above, it is possible to dynamically incorporate a new program in a currently running process. That is, a program component can be replaced without stopping transaction in the currently running process.

On the other hand, the feature of Apfel is that a new program is transferred to a user's computer from a server via a network and is installed based on the determination of the user. That is to say, a program at a user's computer. Therefore, Apfel is essentially different from the dynamic linkage of a program that a new program is loaded while another program is running, as described in the present claimed invention.

Because of applicant's unique combination of claimed features, the dynamic linkage makes it possible to replace programs without service interruption. It is respectfully submitted that this feature is not disclosed by Apfel.

Difference between claims 8, 10, 21, 28 and Apfel

Dependent claim 8 is directed to a management apparatus. Independent claim 10 is a program execution apparatus with similar distinguishing features as recited in claim 1. Impendent claim 21 is a method claim with similar distinguishing features as recited in claim 1. Claim 28 is directed to a method depending to claim 21. For at least the reasons set forth above, it is respectfully submitted that Apfel is different from the claimed inventions and does not sustain the section 102 rejection of claims 8, 10, 21 and 28 due to at least the reasons set forth above with respect to 1.

Boutcher (6,493,768)

Claims 11, 18-20 and 30 are rejected under 35 U.S.C. 102(e) as being anticipated by Boutcher. Claims 19 and 30 are cancelled obviating the rejection.

Summary of Boutcher

Boutcher cited against independent claims 11 and 20 discloses a distributed computer system which includes an upgraded version of a remote procedure interface defining remote procedures and a mapping indication associated with each of the remote procedures. The invention of Boutcher involves application of the remote procedure control typified by Remote Procedure Calls (RPC).

RPC is typically defined using a remote procedure interface. RPC is a transmission method in which a request for a certain operation to a remote computer is considered as a procedure call. Conventionally, assuming that a client computer process supports a first version of a remote procedure and a server computer process supports a second version of the remote procedure, when the client executes the remote procedures to the server, if the server does not support the first version of the remote procedure, the server executes only the second (new) version of the remote procedure. Therefore, it is impossible to establish communication between the client and the server.

In this case, according to Boutcher, the client computer process includes a mapper that maps the first version of the remote procedure to the second version of the remote procedure. Receiving the request for the first version of the remote procedure, the server computer process recognizes that the first version and second version are mapped, and executes the second version of the remote procedure.

Regarding claim 11 of applicant's claimed invention

A client apparatus includes a client stub processing unit which transfers a first message which contains version information indicating a program version. A server apparatus includes a distributed object execution control unit which dynamically links programs corresponding to the version information contained in the first message of which skeleton processing is executed by the server skeleton processing unit, to the memory, so as to enable execution of at least one function in the programs.

Difference between claim 11 and Boutcher

In the Office Action it is asserted this is the same as the plurality of mass storage devices described col. 4, line 54. However there is no suggestion in Boutcher of the dynamically linking to the memory, programs corresponding to the version information contained in the first message of which skeleton processing is executed by the server skeleton processing unit so as to enable execution of at least one function in the programs.

As stated above, as to the remote procedure between the client and the server, Boutcher provides a version map that corresponds versions in the format of remote procedure demand, and therefore makes it possible that the client and the server communicates with each other even if the versions that the client and the server supports are different from each other.

However because of applicant's claimed invention, new software program components can be dynamically loaded, without stopping transactions in a client server system in which distributed object computing is performed. It is respectfully submitted that the claimed invention does not relate to the remote procedure control described in the Boutcher invention.

Independent claim 20 is a server apparatus with similar distinguishing features as recited in claim 11. Consequently, it is submitted that the independent claims 11 and 20 are not anticipated by Boutcher and the rejection should be withdrawn.

Claim 18 is directed to a management server depending from claim 11. For at least the reasons set forth above, it is respectfully submitted that Boutcher is different from the claimed inventions and does not sustain the section 102 rejection of claim 18 due to at least the dependency upon claim 11.

Regarding rejection under 35 U.S.C. § 103

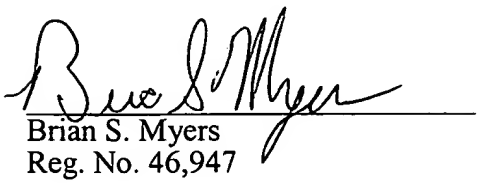
Claims 2, 3, 6, 7, 22, 23, 26 and 27 are rejected under 35 U.S.C. 103 as being unpatentable over Apfel in view of Hunt (6,381,735). Claims 4, 5, 24 and 25 are rejected under 35 U.S.C. 103 as being unpatentable over Apfel in view of Hapner et al. (5,940,827). Claims 12, 13, 16 and 17 are rejected under 35 U.S.C. 103 as being unpatentable over Boutcher (6,493,768) in view of Hunt (6,381,735). Claims 14 and 15 are rejected under 35 U.S.C. 103 as being unpatentable over Boutcher in view of Hapner et al.

For at least the reasons set forth above independent claims 1, 10 and 21 are different from Apfel and independent claims 11 and 20 are different from Boutcher. Therefore, it is respectfully submitted that the dependent claims depending from them are not obvious with the combination of other references with Apfel or Boutcher because such a combination of references is missing at least the distinguishing features as outlined above.

In view of the remarks set forth above, this application is in condition for allowance which action is respectfully requested. However, if for any reason the Examiner should consider this application not to be in condition for allowance, the Examiner is respectfully requested to telephone the undersigned attorney at the number listed below prior to issuing a further Action.

Any fee due with this paper may be charged to Deposit Account No. 50-1290.

Respectfully submitted,



Brian S. Myers
Reg. No. 46,947

CUSTOMER NUMBER 026304

Katten Muchin Zavis Rosenman
Telephone: (212) 940-8703
Fax: (212) 940-8986
Docket No.: FUJR 18.034 (100794-11554)
BSM:fd